Stefano Mazzilli Daechsel

16 Computation

Abstract: This chapter provides an introduction to the computational logics and technologies that are increasingly (re)structuring our societies and the field of criminology. I start by providing a general overview of computation, computational theory, and the technological innovations that underpin contemporary general-purpose computers. I then critically examine ongoing efforts to integrate computational methods into criminological research under the banner of computational criminology.

Keywords: computation, computational theory, computational criminology

Introduction

If digital criminology means to take seriously the "technosocial nature of contemporary social and political life" (Powell et al., 2018: 4), the logics and material components of digital technologies ought to be treated as socially relevant forces. The ubiquity of computational devices in our societies dictates that digital criminologists should have at least a basic understanding of computation and computational theory in order to be attuned to their (re)structuring roles wherever computers are used, including in criminology itself. To this end, I start by providing a general overview of computation, computational theory, and the technological innovations that underpin contemporary general-purpose computers. I then pivot to critically examine ongoing efforts to integrate computational methods into criminological research under the banner of computational criminology.

Computation, computational theory and the general-purpose computer

At a basic level, computation refers to any process that acts on an input to produce an output. Most computation today is digital and electrical (think laptops and smartphones; see Digital by Wernimont) but computational technologies, such as the abacus,

Disclaimers and acknowledgements: Grant numbers and/or funding information: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 947681).

 https://doi.org/10.1515/9783111062037-017

have been around for millennia. Until about the mid-20th century, the most sophisticated computers were highly elaborate constellations of mechanical parts that could perform advanced calculations but were prone to deterioration and breakdown. Today, quantum and molecular computing are testing new models and materials for computation that could result in entirely new applications and capabilities.

Underpinning this great variety of computational technologies is computational theory, a field that has its origins in the discipline of mathematics. Just as computers, broadly defined, are thousands of years old, "the notions of computability and computable functions go back a long time" (Fernandez, 2009: 1), certainly as far back as the ancient Greeks and Egyptians. While figures such as Charles Babbage, Ada Lovelace, and George Boole are often cited as progenitors of modern computing, modern computational theory emerged from efforts in the 1930s by the likes of Alan Turing and Alonzo Church to respond to a series of mathematical problems first posed by David Hilbert at the International Congress of Mathematicians in 1900.

Turing's 1937 article, 'On Computable Numbers, with an Application to the *Entscheidungsproblem,*' describes an abstract computing machine that is composed of an infinite tape divided into squares, "each capable of bearing a 'symbol'" (Turing, 1937: 231), a head that can move along the tape from square to square, and a control unit that dictates how the head should behave given the content of the square it finds itself on. This *Turing machine*, as it is now known, provides the basic elements of the general-purpose computer—a memory unit (the tape), a central processing unit (the head), and a control unit—but Turing's immediate aim in positing this abstract machine was to determine whether Hilbert's *Entscheidungsproblem*, or decision problem, was a computable function. Turing asserts that a function is only computable if it allows the Turing machine to reach a final state, i. e., a final square on the tape, that provides the function's output or solution. The article concludes that Hilbert's decision problem is not a computable function because the Turing machine never reaches a final state but goes on moving from square to square on the tape forever.

Having a working understanding of Turing machines offers some important insights into the nature of computation and computational logics. Turing machines are abstract and thus not bounded by the strictures of the physical universe. Although many physical computers resemble Turing machines in their composition, even the most advanced supercomputers do not have an infinite memory unit, nor infinite energy and time. The abstract plane of computational theory often slams into the material and temporal constraints of reality; what is abstractly computable might not be feasibly computed by a physical machine. These limitations are especially important to keep in mind when computers are used to model and simulate technosocial realities that are highly complex and informationally dense.

To get from the Turing machine to the digital general-purpose computer, three other elements needed to fall into place. John von Neumann's 1945 *First Draft Report on the EDVAC* is considered the first description of a general-purpose electronic computer that can concretely implement all the components of a Turing machine. The *von Neumann architecture* detailed in the report is still dominant today in the world of electronic computing. Information, which is physically instantiated in electrical circuits, passes between the memory unit and the central processing unit in accordance with the inputs provided by the user. Contemporary digital computers essentially embody this architecture.

The works of the mathematician and electrical engineer Claude Shannon were yet another major turning point. In 1937, Shannon wrote what is often considered the most influential master's thesis of all time entitled 'A Symbolic Analysis of Relay and Switching Circuits.' His thesis demonstrated an equivalence between electrical circuits and Boolean algebra, "a symbolic method of investigating logical relationships" (Shannon, 1938: 714) between two variables the result of which is either TRUE or FALSE. Given that this binary logic corresponds to electrical circuits that can be in one of two states, namely, ON or OFF, Shannon showed that electrical circuits could be designed to emulate specific logical relationships between two variables, such as 'AND,' 'NOT,' and 'OR.' His work led to the development of electronic logic gates that continue to form the underlying structure of all digital computers. Shannon's 1948 article, 'A Mathematical Theory of Communication,' is an early classic in the field of information theory and, among many other innovations, established the 'bit' (short for binary digit) as the unit for measuring information in computing and telecommunications.

The binary logic of electronic computing reduces all information to sequences of two discrete states in electrical circuits, ON and OFF, represented symbolically by the bits 1 and 0. On the one hand, the ability to encode virtually any kind of content in a sequence of bits is an incredibly powerful innovation that greatly facilitates the storage and transmission of digital information. On the other hand, converting non-digital information—a sound wave for example—into digital information comes with some potential downsides. As the continuous is made discrete, selection and flattening processes are at work that transform some of the original information and exclude the rest. In many circumstances, this trade-off is absolutely worthwhile, but it is crucial to understand that digital information does not capture the fullness of non-digital reality.

The third and final development to consider in the genesis of the general-purpose computer is the invention of the transistor in 1947 by John Bardeen, Walter Brattain, and William Shockley at the same Bell Labs where Shannon spent much of his career. Often considered "the fundamental building blocks of all modern electronic devices" (Konkoli et al., 2018: 156), transistors are semiconductor devices made of silicon (hence the name Silicon Valley) that can switch or amplify electrical signals. The decades-long trend of transistor miniaturization has meant that more and more transistors can be inserted on an integrated circuit, or microchip, thereby boosting a device's computing power without increasing its physical size. In 1965, the engineer and businessman Gordon Moore speculated that the number of transistors on a microchip would double every year, which he revised to every two years in 1975, a prediction that has largely held and earned the name 'Moore's law.' It is thanks to this process of transistor miniaturization that we are now able to walk around with computers in our pockets that are more powerful than supercomputers that once filled entire laboratories.

However, Moore's law will soon be made obsolete by the simple fact that transistors cannot be miniaturized past the point at which the classical laws of physics give way to quantum mechanics. This real limit on transistor miniaturization, coupled with the immense energy cost of conventional computing, has led to a lot of research on alternatives to silicon-based computing. Quantum computing and molecular computing have received the lion's share of interest and funding, but neither has proven itself capable of exhibiting the versatility and reliability of conventional computers. To date, it appears that these unconventional computing systems will supplement rather than replace conventional computers, finding their true utility in specific tasks.[1]

Criminology and computational power

Part of digital criminology's mandate is not only to understand the impact of computational logics and technologies on crime and crime control, but also on criminology itself. As a field, criminology has long embraced the use of computational methods to store and sort empirical data, such as crime statistics, survey findings, or other datasets (see Datafication by Chan). The larger these datasets have become, the more computer programs are called upon to perform analytical tasks "traditionally undertaken by social scientists" (Williams et al., 2017: 337). This has contributed to the emergence of what several criminologists have dubbed "computational criminology" (Berk, 2008; Williams et al., 2017; Campedelli, 2022; Steinmetz, 2023). Here, however, we must be careful to differentiate between two divergent approaches working under this banner.

On the one hand, there are researchers for whom computational criminology is an interdisciplinary methodology that engages with big data, particularly social media data (see Social Media by Twigt), to help address criminological questions (Williams and Burnap, 2016; Williams et al., 2017; Wiliams et al., 2020). This version of computational criminology sees potential in online data "to complement and augment conventional curated data" (Williams et al., 2017: 321). Computational methods are used to analyze and correlate online data with other datasets, all with the understanding that online data, particularly geolocated and time-stamped data, can provide new insights into long-standing criminological problems. As an example, Williams and Burnap (2016: 217) provide an "analysis of social media data using advanced computing techniques to answer a classic criminological question on social reactions to criminal events of national interest," which in their case was the Woolwich terrorist attack of 2013.

On the other hand, computational criminology is often associated with the development and application of computer simulations for criminological research (Berk, 2008; Malleson et al., 2010; Berk, 2013; Birks, 2018; Groff et al., 2019; Campedelli,

1 This swift overview of the theoretical and technological innovations that underpin contemporary computers should be coupled with the social history of computing and the computerization of society (see Mahoney, 2005, 2011; Campbell-Kelly and Aspray, 2004).

2022). Malleson et al. (2010), for instance, outline a computer model to simulate the occurrence of residential burglaries in different neighborhood constellations. The authors employ an agent-based model that simulates the behavior of artificial agents ('citizens' and 'potential burglars') that are assigned specific rules of interaction and certain drives, such as the need to sleep and the need to generate wealth (Malleson et al., 2010: 239). In this variation of computational criminology, abstraction and highly reductive reasoning are used to transform complex social environments and individuals into computable functions.

Advocates of computer simulations in criminology note that processes of abstraction are inherent to the formulation of any theory (Birks, 2018). Indeed, all theorists abstract the dynamics and variables they deem salient to a given problem from the complexity of social life. The difference, however, lies in the degrees and layers of abstraction needed to reduce social complexity to a computable model that simulates both agents and their environments. Whereas the first approach to computational criminology aims to develop methodological tools to better explore the significance of digital data that are already circulating and acting within society, the second form of computational criminology largely supplants the social with models and simulations that, no matter how sophisticated, are reductive abstractions that must comply with the logics and material constraints (time, energy, memory, computing power) of existing technologies. Even if one assumes "an infinite amount of computational power at [one's] disposal, it remains very challenging to try to accurately model a virtual society in all its facets and dimensions" (Campedelli, 2022: 63).

Conclusion

My aim in this chapter was to provide a general introduction to the computational logics and technologies that are increasingly (re)structuring our societies and the field of criminology itself. As digital criminologists, we should challenge strict binary oppositions between technology and society and advance a technosocial approach that treats technological logics and systems as social forces that must be understood on their own terms. Thus, here are the chapter's main takeaways:

– The history of computation reveals a process by which abstract machines, such as the Turing machine, and digital logics, such as Boolean algebra, gradually materialized into the general-purpose computers that we are all familiar with today. Unlike abstract machines, however, material computers are constrained by factors such as time, energy, memory, and computing power. Because of these constraints, computer scientists are exploring new computational models and materials, such as quantum computing and molecular computing.
– Computational logics are incredibly powerful and versatile when the information in question is digital (bits of 1s and 0s). However, translating non-digital information into digital information always involves processes of abstraction and simplification (see Translation by Wilson-Kovacs). The original signal is never fully cap-

tured by its digital counterpart. Digital criminologists should take account of these constraints when they study computers as knowledge-making devices or as a part of social practices.

– Criminologists have long used computers as part of their research, but there are ongoing efforts to integrate more sophisticated computational methods into criminology. The rise of computational criminology reflects this trend, although we must be attentive to the different approaches working under the same banner.
– While some computational criminologists are developing computational methods to integrate existing digital information, such as social media content, into criminological research, others are constructing computer models to simulate highly complex social dynamics, models that invariably reduce social reality to oversimplified (but computable) abstractions.

Suggested reading

Berk, R. (2013). Algorithmic criminology. *Security Informatics*, 2(5), 1 – 14.
Mahoney, M. S. (2005). The histories of computing(s). *Interdisciplinary Science Reviews*, 30(12), 119 – 135.
Steinmetz, K. (2023). The mundanity of computational criminology. In *Against Cybercrime* (pp. 155 – 187). London: Routledge.
Williams, M., Burnap, P., & Sloan, L. (2017). Crime sensing with big data: The affordances and limitations of using open-source communications to estimate crime patterns. *British Journal of Criminology*, 57, 320 – 340.

References

Berk, R. (2008). How you can tell if the simulations in computational criminology are any good. *Journal of Experimental Criminology*, 4, 289 – 308.
Berk, R. (2013). Algorithmic criminology. *Security Informatics*, 2(5), 1 – 14.
Birks, D. (2018). Computer simulations: Agent-focused environmental criminology. In G. Bruinsma & S. Johnson (eds.), *The Oxford Handbook of Environmental Criminology* (pp. 311 – 340). Oxford: Oxford University Press.
Campbell-Kelly, M., & Aspray, W. (2004). *Computer: A History of the Information Machine*, Boulder, CO: Westview Press.
Campedelli, G. M. (2022). *Machine Learning for Criminology and Crime Research: At the Crossroads.* London: Routledge.
Fernandez, M. (2009). *Models of Computation: An Introduction to Computability Theory.* London: Springer.
Groff, E., Johnson, S., & Thornton, A. (2019). State of the art in agent-based modeling of urban crime: An overview. *Journal of Quantitative Criminology*, 35, 155 – 193.
Konkoli, Z., Stepney, S., Broersma, H., Dini, P., Nehaniv, C., & Nichele, S. (2018) Philosophy of computation. In S. Stepney, S. Rasmussen, & M. Amos (eds.), *Computational Matter* (pp. 153 – 184). London: Springer.
Mahoney, M. S. (2005). The histories of computing(s). *Interdisciplinary Science Reviews*, 30(12), 119 – 135.

Mahoney, M. S. (2011). *Histories of Computing.* Cambridge, MA: Harvard University Press.
Malleson, N., Heppenstall, A., & See, L. (2010). Crime reduction through simulation: An agent-based model of burglary. *Computers, Environment and Urban Systems*, 34, 236 – 250.
Powell, A., Stratton, G., and Cameron, R. (2018). *Digital Criminology: Crime and Justice in Digital Society.* London: Routledge.
Shannon, C. (1938). A symbolic analysis of relay and switching circuits. *Transactions of the American Institute of Electrical Engineers*, 57(12), 713 – 723.
Shannon, C. (2001). A mathematical theory of communication *Mobile Computing and Communications Review*, 5(1), 3 – 55.
Steinmetz, K. (2023). *Against Cybercrime.* London: Routledge.
Turing, A. (1937). Computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1), 230 – 265.
Von Neumann, J. (1993). First draft of a report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4), 27 – 75.
Williams, M., & Burnap, P. (2016). Cyberhate on social media in the aftermath of Woolwich: A case study in computational criminology and big data. *British Journal of Criminology*, 56, 211 – 238.
Williams, M., Burnap, P., & Sloan, L. (2017). Crime sensing with big data: The affordances and limitations of using open-source communications to estimate crime patterns. *British Journal of Criminology*, 57, 320 – 340.
Williams, M., Burnap, P., Javed, A., Lui, H., & Ozalp, S. (2020). Hate in the machine: Anti-black and anti-Muslim social media posts as predictors of offline racially and religiously aggravated crime. *British Journal of Criminology*, 60, 93 – 117.